

高等院校计算机实验与实践系列示范教材

基于FPGA的硬件系统设计 实验与实践教程

姚爱红 张国印 武俊鹏 主编



清华大学出版社

高等院校计算机实验与实践系列示范教材

基于FPGA的硬件系统设计 实验与实践教程

清华大学出版社
北京

内 容 简 介

本书介绍基于FPGA的数字系统设计方法,在基本的功能部件设计基础上,设计微处理器及单芯片系统。本书不仅介绍通用的数字电路和数字系统的设计方法,并对计算机硬件系统的组织进行深入分析。通过运算电路设计、状态机设计、RISC模型机设计等实验用例的训练,学生可以建立计算机的整机概念,了解数据在计算机中的表示、传送、处理以及控制信息是如何完成对计算机系统的控制。

本书取材新颖,采用实例教学的组织形式,内容由浅入深,循序渐进。书中给出了大量设计实例及扩展方案,不仅可以作为教学内容进行学习,部分内容还具有工程实践价值。本书可作为高等院校计算机类、电子类和自动化类等有关专业的教材和参考书,也可供有关专业工程技术人员参考。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

基于FPGA的硬件系统设计实验与实践教程/姚爱红,张国印,武俊鹏主编.--北京:清华大学出版社,2011.6

(高等院校计算机实验与实践系列示范教材)

ISBN 978-7-302-24537-7

I. ①基… II. ①姚… ②张… ③武… III. ①可编程序逻辑器件—系统设计—高等学校—教材 IV. ①TP332.1

中国版本图书馆CIP数据核字(2011)第009658号

责任编辑:索梅薛阳

责任校对:白蕾

责任印制:何芊

出版发行:清华大学出版社

地 址:北京清华大学学研大厦A座

<http://www.tup.com.cn>

邮 编:100084

社 总 机:010-62770175

邮 购:010-62786544

投稿与读者服务:010-62795954,jsjic@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015,zhiliang@tup.tsinghua.edu.cn

印 刷 者:北京富博印刷有限公司

装 订 者:北京市密云县京文制本装订厂

经 销:全国新华书店

开 本:185×260 印 张:19 字 数:463千字

版 次:2011年6月第1版 印 次:2011年6月第1次印刷

印 数:1~3000

定 价:29.00元

产品编号:025293-01

出版说明

当前,重视实验与实践教育是各国高等教育界的发展潮流,我国与国外教学工作的差距也主要表现在实践教学环节上。面对新的形势和新的挑战,完善实验与实践教育体系成为一种必然。为了培养具有高质量、高素质、高实践能力和高创新能力的人才,全国很多高等院校在实验与实践教学方面进行了大力改革,在实验与实践教学内容、教学方法、教学体系、实验室建设等方面积累了丰富的宝贵经验,起到了教学示范作用。

实验与实践性教学与理论教学是相辅相成的,具有同等重要的地位。它是在开放教育的基础上,为配合理论教学、培养学生分析问题和解决问题的能力以及加强训练学生专业实践能力而设置的教学环节;对于完成教学计划、落实教学大纲,确保教学质量,培养学生分析问题、解决问题的能力和实际操作技能更具有特别重要的意义。同时,实践教学也是培养应用型人才的重要途径,实践教学质量的好坏,实际上也决定了应用型人才培养质量的高低。因此,加强实践教学环节,提高实践教学质量,对培养高质量的应用型人才至关重要。

近年来,教育部把实验与实践教学作为对高等院校教学工作评估的关键性指标。2005年1月,在教育部下发的《关于进一步加强高等学校本科教学工作的若干意见》中明确指出:“高等学校要强化实践育人的意识,区别不同学科对实践教学的要求,合理制定实践教学方案,完善实践教学体系。要切实加强实验、实习、社会实践、毕业设计(论文)等实践教学环节,保障各环节的时间和效果,不得降低要求。”“要不断改革实践教学内容,改进实践教学方法,通过政策引导,吸引高水平教师从事实践环节教学工作。要加强产学研合作教育,充分利用国内外资源,不断拓展校际之间、校企之间、高校与科研院所之间的合作,加强各种形式的实践教学基地和实验室建设。”

为了配合开展实践教学及适应教学改革的需要,我们在全中国各高等院校精心挖掘和遴选了一批在计算机实验与实践教学方面潜心研究并取得了富有特色、值得推广的教学成果的作者,把他们多年积累的教学经验编写成教材,为开展实践教学的学校起一个抛砖引玉的示范作用。

为了保证出版质量,本套教材中的每本书都经过编委会委员的精心筛选和

严格评审,坚持宁缺毋滥的原则,力争把每本书都做成精品。同时,为了能够让更多、更好的实践教学成果应用于社会和各高等院校,我们热切期望在这方面有经验和成果的教师能够加入到本套丛书的编写队伍中,为实践教学的发展和取得成效做出贡献;也衷心地期望广大读者对本套教材提出宝贵意见,以便我们更好地为读者服务。

清华大学出版社

联系人:索梅 suom@tup.tsinghua.edu.cn

1. 写作背景

目前,关于我国高等教育计算机专业一级学科和专业方向设置问题受到非常大的关注。原有的计算机科学与技术一级学科现在逐渐分化为计算机科学、计算机工程和软件工程 3 个学科,同时保密与信息安全专业方向也在积极地进行准备。对于计算机专业的本科生教育而言,其基本的培养方案、课程设置和教学大纲都需要根据新的形势进行变革。

传统的计算机专业课程体系中与计算机硬件设计相关的课程是“数字逻辑”、“计算机组成原理”和“计算机系统结构”,每一门课程都辅助有相应的实践学时。实践环节的设置不仅起到加深学生理解理论课所学知识的作用,还有助于培养学生将理论与实践相联系并解决实际问题的能力,对于实现当前的高等教育改革目标,提高毕业生综合素质具有重要的意义。但是,受实验设备所限,各课程的实验环节分别在数字逻辑实验箱、计算机部件实验平台和 EDA 实验开发平台上进行,缺乏连贯性和整体性。

随着现场可编程逻辑器件的广泛应用,以 FPGA 为核心器件的实验教学平台日益成熟,采用 FPGA 芯片实现数字逻辑、计算机部件和计算机组成及系统设计实验已经成为新的趋势。采用 FPGA 芯片实现硬件设计实验,开发速度快、方便、可靠,并且由于 FPGA 芯片可反复编程,因此几乎没有器件损耗,大大降低了实验室的维护成本。另一方面,随着可编程逻辑器件集成度的迅速提高和嵌入式系统应用的日益广泛,基于 FPGA 的 SoPC 系统设计技术已经在通信、工业控制等许多领域得到实际应用。在计算机专业硬件课程的实验教学环节中引入 SoPC 相关的内容,渗透片上系统软硬件协同设计的思想,对于提高学生的实际动手能力和就业竞争力都有非常大的帮助。

2. 本书特点

本书兼顾高等学校理论教学需要与培养学生实践能力的需求,借鉴国外名校在计算机科学与工程专业课程设置及相关课程内容的安排,组织本教程的相关理论知识及实验用例设计,力争理论详尽、用例科学、指导到位。配合高等学校的“数字逻辑”、“计算机组成原理”、“计算机系统结构”等课程的实践教学环节,突出实用性。所有实验可操作性强,与实践结合紧密。本书不仅介绍基于 FPGA 的 EDA 设计方法,更着眼于介绍在基本的数字逻辑功能部件的基础上如何构造微处理器,即本教材着眼于计算机的硬件系统组织,而不仅仅是介绍通用

的数字电路和数字系统的设计方法。通过浮点运算电路设计、状态机设计、RISC 模型机部分模块的设计等实验用例的训练,学生可以建立计算机的整机概念,了解数据在计算机中的表示、传送、处理以及控制信息是如何完成对计算机系统进行控制的。

本书可作为计算机科学与技术专业及相关专业“数字逻辑实验”、“EDA 设计实验”、“计算机部件实验”、“计算机组成实验”等课程的教材,书中的全部实验示例都经过精心的设计和完全的调试,可以放心使用。除数字钟、VGA 接口控制器和 SoPC 系统设计这 3 章的实验结果需要下载到目标平台上以外,其他实验均可直接在仿真开发环境中进行,方便可靠。此外,与平台相关的实验内容也只需少量修改(主要是修改引脚约束文件),就可以移植到不同的实验平台上运行。

3. 内容安排

本书的内容安排如下:

- 第 1 章介绍可编程逻辑器件的基本概念、发展历程以及分类方法和开发流程,并介绍 FPGA 的发展及主要厂商和产品的概况。
- 第 2 章简要介绍 Verilog HDL 的基本语法知识、进行硬件描述的一般结构及使用硬件描述语言进行电路设计的 3 种描述方式(结构描述、数据流描述和行为级描述)。
- 第 3 章介绍进行 FPGA 开发所需的工具软件、目前国内高校常用的 FPGA 实验平台以及一些常用仪器仪表的使用方法,并以实验 0 的形式详细介绍基于 FPGA 的硬件设计过程。
- 第 4~5 章以常用的组合逻辑电路和几种常见的同步触发器电路为例,分别详细描述采用原理图输入法和硬件描述语言方法设计组合逻辑、时序逻辑及功能仿真的方法。
- 第 6 章简要介绍 FSM 中状态的描述方法、不同类型 FSM 的设计方法、FSM 的复位和毛刺问题等。通过乘法状态机、序列检测器和交通灯控制器 3 个示例说明 FSM 的设计方法。
- 第 7 章介绍定点加法器和浮点加法器的设计,详细描述了一个 32 位浮点加法器和一个简单算术逻辑运算单元的设计实现过程。
- 第 8 章介绍基本乘法器和除法器的原理与设计方法,以及两种快速乘法器设计技术。
- 第 9 章讨论对存储器的建模,介绍利用 Verilog HDL 对只读存储器和随机存取存储器进行建模,以及利用 IP Core 生成工具 MegaWizard Plug-In Manager 生成存储器核的方法。
- 第 10 章通过详细设计一个教学用模型微处理器 HEU-R1,介绍处理器设计的基本方法和步骤。
- 第 11 章详细描述了利用 Verilog HDL 语言设计基于 FPGA 的数字电子时钟的过程,而且所设计的电路具有工程实用性。
- 第 12 章介绍 VGA 接口的基本原理和控制方式,通过设计并实现了在 CRT 显示器上显示彩色条块、圆环的示例来说明基于 FPGA 的 VGA 接口设计技术,并在 GW48-SoPC 实验台上进行实验。
- 第 13 章阐述 FIR 滤波器的基本原理、结构以及窗函数法设计 FIR 滤波器的过程,并

详细描述在 FPGA 上实现 FIR 滤波的方法及验证过程。

- 第 14 章介绍利用 Altera 公司提供的嵌入式微处理器核 Nios 实现 SoPC 系统的基本方法,以一个跑马灯控制器的软硬件设计为例对 SoPC 系统设计流程进行详细描述。

4. 致谢

首先感谢哈尔滨工程大学计算机科学与技术学院嵌入式系统研究室和计算机实验教学中心的各位老师和研究生,感谢他们的大力支持和热情帮助。以下研究生参与了本书实验示例代码的编写和调试以及原始资料的翻译和整理工作:康萌萌、刘铭、张智钧、孙盟哲、王春艳、关琳、颜丽、吴剑、李玲、谢争、杜小雪、菅庆峰、张明涛等,感谢他们付出的辛勤劳动。感谢刘书勇老师和付晓晶老师的热情帮助,他们为编者提供了实验平台及相关资料。

感谢评阅专家对本书提出的宝贵修改意见,他们的劳动对完善和提高全书质量起到了关键的作用。

编者虽然从事计算机硬件课程教学多年,但是由于水平所限,书中难免存在缺点和错误,希望各位读者提出宝贵意见。编者的联系方式为: yaoaihong@hrbeu.edu.cn。

编 者

2011 年 2 月

第 1 章 可编程逻辑器件简介	1
1.1 可编程逻辑器件概述	1
1.1.1 可编程逻辑器件的发展历程	2
1.1.2 可编程逻辑器件的分类方法	5
1.2 可编程逻辑器件的设计流程	6
1.3 FPGA 发展概况	9
1.3.1 FPGA 的主要优势与发展前景	9
1.3.2 主流 FPGA 产品及供应商简介	9
习题 1	11
第 2 章 Verilog HDL 基础	12
2.1 前言	12
2.2 程序示例	12
2.3 模块	14
2.3.1 模块的结构	14
2.3.2 模块的实例化	15
2.4 变量和信号的类型	16
2.5 Verilog HDL 表达式	17
2.5.1 常量	17
2.5.2 操作符	18
2.6 Verilog HDL 的主要功能语句	19
2.6.1 Verilog HDL 对硬件的描述方式	19
2.6.2 数据流描述	20
2.6.3 行为描述——过程块	20
2.7 其他语法规则说明	24
2.7.1 标识符命名原则	24
2.7.2 标点的使用	25
2.7.3 注释	25
2.7.4 转义字符	25

2.7.5 编译命令	25
2.7.6 参数	26
2.8 示例程序分析	26
2.9 推荐阅读	27
习题 2	27
第 3 章 实验环境介绍	28
3.1 EDA 软件工具介绍	28
3.1.1 集成的 FPGA 开发环境	28
3.1.2 ModelSim 介绍	30
3.1.3 Synplify 简介	30
3.2 FPGA 典型实验开发平台简介	31
3.2.1 康芯 GW48-SoPC 实验台	31
3.2.2 Xilinx XUP Spartan 板	31
3.3 实验仪器的使用方法	32
3.3.1 函数信号发生器	32
3.3.2 数字存储示波器	32
3.3.3 逻辑分析仪	33
3.4 熟悉实验环境	33
3.4.1 实验目的	34
3.4.2 实验内容	34
3.4.3 实验步骤	34
习题 3	45
第 4 章 基本组合逻辑电路设计	47
4.1 组合逻辑电路基础知识	47
4.1.1 组合逻辑电路的分析方法	47
4.1.2 组合逻辑电路分析举例	48
4.1.3 组合逻辑电路的设计方法	48
4.2 数据比较器	49
4.2.1 数据比较器的功能	49
4.2.2 比较器电路的设计	49
4.3 数据选择器	53
4.3.1 四选一数据选择器	53
4.3.2 四选一数据选择器的设计	53
4.3.3 数据选择器的应用	58
4.4 二进制加法器	59
4.4.1 半加器	59
4.4.2 全加器	61

4.5 编码/译码器	64
4.5.1 BCD 码编码器	64
4.5.2 BCD 码译码器	67
实验 4-1 用原理图输入法设计四位加法器	70
实验 4-2 数码显示译码器	70
习题 4	72
第 5 章 基本时序逻辑设计	73
5.1 时序逻辑电路的基础知识	73
5.2 触发器	74
5.2.1 RS 触发器	74
5.2.2 D 触发器	76
5.2.3 JK 触发器与 T 触发器	77
5.3 时序逻辑电路的分析方法	79
5.3.1 同步时序电路的分析方法	79
5.3.2 异步时序电路的分析方法	80
5.4 常见的时序逻辑电路设计	82
5.4.1 移位寄存器	82
5.4.2 计数器	85
5.4.3 分频器	87
5.4.4 顺序脉冲发生器	91
5.4.5 阶乘运算器	94
实验 5-1 可预置的加减计数器实验	96
实验 5-2 扭环形计数器	97
习题 5	98
第 6 章 有限状态机设计	99
6.1 状态的描述	99
6.1.1 整数编码状态	99
6.1.2 parameter 语句声明状态	100
6.1.3 `define 编译引导语句	100
6.2 FSM 的设计方法	101
6.2.1 Moore 型 FSM 的设计	101
6.2.2 Mealy 型 FSM 的设计	104
6.2.3 混合型 FSM 的设计	107
6.3 FSM 的复位和毛刺问题	111
6.4 FSM 设计示例	112
6.4.1 乘法器建模	112
6.4.2 序列检测器的设计	117

6.4.3 交通灯控制器的设计	119
实验 6-1 设计序列检测器	123
习题 6	124
第 7 章 加法器设计	126
7.1 定点加法器	126
7.1.1 进位链结构	126
7.1.2 串行进位	127
7.1.3 并行进位	129
7.2 浮点加法器	131
7.2.1 规格化浮点数加减运算基本原理	132
7.2.2 浮点加法器的设计	133
7.3 运算器(ALU)的设计	139
实验 7-1 8 位加法器的设计	143
实验 7-2 16 位超前进位加法器	144
习题 7	145
第 8 章 乘、除法器的设计	146
8.1 常用的机器数编码格式	146
8.2 定点乘法器原理及实现	147
8.2.1 原码一位乘算法及实现	147
8.2.2 补码一位乘算法及实现	152
8.3 定点除法器原理及实现	157
8.3.1 原码不恢复余数除法	158
8.3.2 补码不恢复余数除法	162
8.4 快速乘法器	165
8.4.1 修正布斯算法	165
8.4.2 华莱士树结构	166
实验 8-1 原码两位乘法器	169
实验 8-2 补码两位乘法器	170
习题 8	171
第 9 章 存储器建模	172
9.1 只读存储器 ROM 的建模	172
9.1.1 ROM 的基本结构	172
9.1.2 ROM 的建模	173
9.1.3 ROM 的仿真测试	174
9.2 随机存储器 RAM 的建模	175
9.2.1 RAM 的基本结构	175

9.2.2	RAM 的建模	175
9.2.3	RAM 的仿真测试	179
9.3	利用 IP Core 工具生成 ROM 和 RAM	181
实验 9-1	利用 SRAM 设计并实现 FIFO	184
习题 9		185
第 10 章	CPU 的设计	186
10.1	CPU 的基本组成	186
10.1.1	控制部件	186
10.1.2	运算部件	188
10.1.3	寄存器组	188
10.2	CPU 设计的一般过程	188
10.3	HEU-R1 处理器指令集的设计	189
10.3.1	指令格式	189
10.3.2	指令集的设计	190
10.4	HEU-R1 内部数据通路的设计	191
10.5	时序系统的设计	192
10.6	HEU-R1 各功能模块的设计	192
10.6.1	指令译码模块的设计	192
10.6.2	立即数生成模块	194
10.6.3	分支处理模块	194
10.6.4	地址生成模块	195
10.6.5	算术逻辑单元模块	196
10.6.6	寄存器组模块	197
10.6.7	CPU 模块	198
10.7	仿真验证及结果	201
10.7.1	外围模块建模	201
10.7.2	系统复位	204
10.7.3	功能验证	204
实验 10-1	HEU-R1 处理器核的指令集扩展	207
习题 10		208
第 11 章	数字电子时钟设计	209
11.1	数字钟功能需求说明	209
11.2	实验平台相关电路说明	209
11.2.1	7 段数码管	210
11.2.2	外部按键	210
11.2.3	音频输出	211
11.3	数字钟系统的设计	212

11.4 数字钟各模块的设计	212
11.4.1 时钟分频模块	212
11.4.2 计时模块(包含按键控制)	213
11.4.3 音频输出模块	215
11.5 仿真验证	218
11.6 引脚设置	219
实验 11-1 整点报时闹钟设计	219
习题 11	220
第 12 章 VGA 接口控制器	221
12.1 视频信号原理	221
12.2 数字视频图像的表示	223
12.3 VGA 接口介绍	225
12.4 VGA 信号时序	225
12.5 VGA 接口控制器设计	227
12.5.1 VGASig 模块	228
12.5.2 ColorMap 模块	230
12.5.3 顶层模块	231
12.5.4 功能仿真	233
12.5.5 引脚设置	233
实验 12-1 800×600 分辨率 VGA 接口的设计	234
实验 12-2 VGA 动态图形显示控制	235
习题 12	236
第 13 章 FIR 数字滤波器设计	237
13.1 数字滤波器概述	237
13.2 FIR 滤波器的结构	238
13.3 FDATool 工具使用介绍	240
13.3.1 MATLAB 简介	240
13.3.2 FDATool 设计 FIR 滤波器的参数	240
13.4 窗函数法 FIR 滤波器的设计	242
13.4.1 窗函数的选择	243
13.4.2 窗函数法 FIR 滤波器的设计步骤	243
13.5 FIR 滤波器的 FPGA 实现	244
13.5.1 滤波器系数的量化	244
13.5.2 16 阶 FIR 滤波器的实现	245
13.5.3 在 ModelSim 中加入 Altera 仿真库	250
13.6 FIR 滤波器的仿真验证	251
13.6.1 仿真数据文件的格式	251

13.6.2	测试平台程序的设计	252
13.6.3	仿真结果分析	254
实验 13-1	低通 FIR 滤波器的设计	256
实验 13-2	FIR 滤波器的硬件实现及仿真	258
习题 13		259
第 14 章	基于 Nios 的 SoPC 系统	260
14.1	SoPC 技术概述	260
14.1.1	IP 核与 IP 复用技术	261
14.1.2	片上总线	262
14.2	嵌入式微处理器核介绍	263
14.2.1	Altera Nios II 软核处理器	263
14.2.2	Xilinx MicroBlaze 核	264
14.3	基于 Nios 的 SoPC 系统开发流程	266
14.4	基于 Nios 的跑马灯控制器的设计	267
14.4.1	基本 SoPC 系统硬件结构	267
14.4.2	JTAG UART IP 核	268
14.5	跑马灯控制器的硬件实现	269
14.5.1	新建 SoPC 设计项目	269
14.5.2	各模块的设计	271
14.5.3	存储器地址和 IRQ 分配	275
14.5.4	Nios II 系统生成	275
14.5.5	SoPC 系统生成	276
14.6	跑马灯控制器的软件设计	279
14.6.1	C 源程序输入	280
14.6.2	代码优化	281
14.6.3	程序运行和下载	282
实验 14-1	基于 Nios II 处理器计时器的设计	284
习题 14		285
附录	Verilog HDL 关键字	286
参考文献		287

通过本章的学习,读者将较为系统地了解可编程逻辑器件及 FPGA 器件,并对其在电子系统开发中的作用和意义有一定的认识。

1.1 可编程逻辑器件概述

自 20 世纪 60 年代初诞生以来,集成电路经历了中小规模集成电路、大规模集成电路、超大规模集成电路几个发展阶段,与此同时,数字系统的设计和实现方法也随之发生了巨大的变化。

随着集成电路制造工艺和编程技术的提高,为了改进电路设计离不开制造厂的缺点,从 20 世纪 70 年代末开始,发展了一种称为可编程逻辑器件(PLD, Programmable Logic Device)的半定制芯片。PLD 芯片内的硬件资源和连线资源是由制造厂生产好的,但用户可以借助功能强大的设计自动化软件和编程器,自行在实验室、研究室,甚至车间等生产现场进行设计和编程,实现所希望的数字系统。

近年来,PLD 作为专用集成电路(ASIC, Application Specific Integrated Circuit)的一个重要分支,其制造技术和应用技术都取得了飞速的发展。最初只有与-或单元结构的简单 PLD(SPLD, Simple PLD),随着集成技术的进一步发展,出现了集成规模更大、密度更高的 PLD(CPLD, HDPLD)。HDPLD 又可分为复杂 PLD(Complex PLD)和现场可编程门阵列(FPGA, Field-Programmable Gate Array)两类:在有效扩展 SPLD 和吸取标准单元阵列的构思基础上,构成了 CPLD,它已不再完全局限于由两级与-或电路构成的阵列;在门阵列的基础上发展出 FPGA。随半导体工艺水平的不断提高,电路的集成规模按照摩尔定律成倍增长,在一个芯片上已经可以集成一个独立的系统,完成特定的功能,这就是所谓的片上系统或系统芯片(SoC, System on Chip)。ASIC 和 FPGA 是实现 SoC 的两大途径。FPGA 在可编程能力、设计成本、上市速度方面具有 ASIC 不可比拟的优点,而 ASIC 在价格以及实现系统的某些特定物理特性上又有 FPGA 无法实现的优势,为了解决这方面的矛盾,人们提出了将两种技术创造性地融合在一起,发展出了具有可重配置功能模块的 ASIC 芯片和具有 ASIC 功能模块的 PLD 芯片。PLD 的分类如图 1-1 所示。

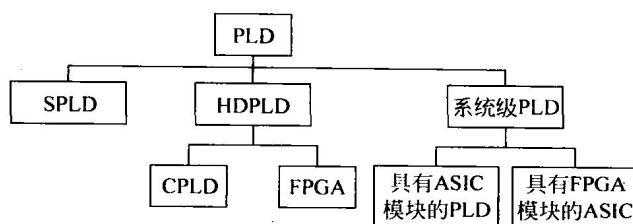


图 1-1 PLD 器件的分类

1.1.1 可编程逻辑器件的发展历程

1. 可编程只读存储器 PROM

从原理上说,可编程逻辑器件的原型就是只读存储器(ROM, Read-only Memory)。从结构上看,ROM 是由一个不可编程的与阵列和一个可编程的或阵列构成的,如图 1-2 所示。

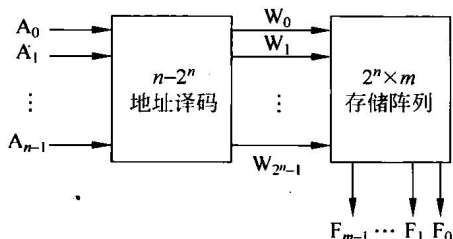


图 1-2 ROM 的基本结构

它包含一个固定的与阵列和一个可编程的或阵列,ROM 中的与阵列是全译码形式,它产生 n 个输入变量的所有最小项,每个输出端通过或阵列将这些最小项有选择地进行或运算,即可实现任何组合逻辑函数。由于与阵列能够产生输入变量的全部最小项,所以用 ROM 实现组合逻辑函数不需要进行逻辑化简。早期的可编程只读存储器(PROM, Programmable ROM)、紫外线可擦除只读存储器

(EPROM, Erasable Programmable ROM)和电可擦除只读存储器(E²PROM, Electrical Erasable Programmable ROM)都属于这种类型。

但随着输入变量数的增加,与阵列的规模会迅速增大,其价格也随之大大提高。而且与阵列越大,译码开关时间就越长,相应的工作速度也越慢。因此,实际上只有规模较小的 PROM 可以有效地实现组合逻辑函数,而大规模的 PROM 价格高,工作速度低,一般只作为存储器使用。由于这种结构上的限制,它们只能完成简单的数字逻辑功能。

2. 可编程逻辑阵列 PLA

真正意义上的 PLD 是 20 世纪 70 年代出现的可编程逻辑阵列(PLA, Programmable Logic Array)。相对于 ROM 中固定的与阵列,PLA 的结构特点是输入逻辑的与阵列和输出逻辑部分的或阵列均可编程。PLA 和 PROM 一样也是组合型逻辑阵列,与 PROM 不同的是,它的两个逻辑阵列均可编程。PLA 的与阵列不再采用全译码的形式,从而减小了阵列规模,集成度相对提高。但是由于在实现函数时,只运用简化后的与或式来实现所需的组合函数,因此 PLA 在编程前必须先进行函数化简。然而,由于 PLA 器件的与、或阵列都可以进行编程,赋予它很大的灵活性的同时,编程难度较大;由于其实际结构简单,器件的引脚有限,无法满足大规模逻辑高度灵活编程的需要,因而无法体现该结构的与或阵列都可编程的优越性;另外,PLA 器件的开发工具应用并不广泛,编程一般由生产厂家完成。因此,实际上 PLA 器件并没有形成产品。